# A Comparison of Front-Ends for Bitstream-Based ASR over IP

Carmen Peláez-Moreno, Ascensión Gallardo-Antolín,
Diego F. Gomez-Cajas and Fernando Díaz-de-María

*Dpto. de Teoría de la Señal y Comunicaciones*
*EPS-Universidad Carlos III de Madrid*
*Avda. de la Universidad, 30, 28911-Leganés (Madrid), SPAIN*
*Phone: +34 91 624 8771*
*Fax: +34 91 624 8749*

**Abstract**

Automatic Speech Recognition (ASR) is called to play a relevant role in the provision of spoken interfaces for IP-based applications. However, as a consequence of the transit of the speech signal over these particular networks, ASR systems need to face two new challenges: the impoverishment of the speech quality due to the compression needed to fit the channel capacity and the inevitable occurrence of packet losses.

In this framework, bitstream-based approaches that obtain the ASR feature vectors directly from the coded bitstream, avoiding the speech decoding process, have been proposed ([4,7,10,15], among others) to improve the robustness of ASR systems. LSP (Line Spectral Pairs) are the preferred set of parameters for the description of the speech spectral envelope in most of the modern speech coders. Nevertheless, LSP have proved to be unsuitable for ASR, and they must be transformed into cepstrum-type parameters. In this paper we comparatively evaluate the robustness of the most significant LSP to cepstrum transformations in a simulated VoIP (Voice over IP) environment which includes two of the most popular codecs used in that network (G.723.1 and G.729) and several network conditions. In particular, we compare 'pseudocepstrum' [9], an approximated but straightforward transformation of LSP into LP cepstral coefficients, with a more computationally demanding but exact one. Our results show that pseudocepstrum is preferable when network conditions are good or computational resources low, while the exact procedure is recommended when network conditions become more adverse.

*Key words:* Robust speech recognition, speech coding, IP networks, coding distortion, packet loss, LSP

*Email address:* carmen, gallardo, dgomez and fdiaz@tsc.uc3m.es (Fernando Díaz-de-María).

## 1 Introduction

As voice transmission over IP networks (VoIP) becomes popular, new voice-enabled services provided through these networks are being developed. Therefore, ASR (Automatic Speech Recognition) is called to play an important role in the provision of user-friendly spoken interfaces for these services. However, under those circumstances, two VoIP-specific problems emerge: first, the scarcity of bandwidth makes the use low-to-medium-rate speech coders necessary and, consequently, coding distortion reduces the recognizers accuracy [5,12]; and second, packet losses, severely affect ASR performance [15].

Recent papers ([3–5,7,10], among others) have established that more robust parameterizations can be obtained by transforming some of the parameters sent by the coder, instead of decoding the speech signal and using a conventional ASR front-end. This means that selecting just the necessary information from the bitstream is better than extracting it from the decoded waveform.

The motivations are the following: first, the avoidance (except for quantization) of the encoding-decoding distortion; second, the possibility of selecting just the relevant information for recognition from the bitstream, therefore minimizing the likelihood that the feature extraction process be influenced by irrelevant (from the ASR point of view) or erroneous information (due to channel distortions); and third, the error recovery mechanisms provided by the standard coders can also be improved, adapting them to the ASR problem. This can be achieved by relaxing the restrictions posed by the coding procedures such as maximum delays or light-weight interpolation methods.

Most of the modern speech coders (G.723.1, G.729, and the new AMR -Adaptive Multi Rate- set of coders, for example) employ LSP (Line Spectral Pairs, also called LSF -Line Spectral Frequencies-) parameters for the coding of the speech spectral envelope [11]. There are a number of reasons that motivate this choice: first, they are highly predictable (they give smooth frame to frame transitions); second, their interpretation as frequencies eases the integration of auditory-related concepts; finally, they offer the possibility of performing a straight-forward stability check. However, the use of LSP as feature vectors has proved to be unsuitable for current ASR systems [4]. Therefore, they must be transformed into MFCC-type (Mel Frequency Cepstral Coefficients) parameters, which nowadays are still the most successful parameters for ASR.

Since bitstream-based ASR front-ends turn out to be more robust for dealing with compressed speech, and current coders use LSPs parameters for representing the speech spectral envelope, the study of the robustness of the transformation methods for obtaining MFCC-type parameters from LSPs becomes relevant. Thus, in this paper we conduct a comparative evaluation of alter-

native computation methods to obtain mel-scaled LPCC (Linear Prediction Cepstral Coefficients), i.e., the calculation of MFCC from LP-based parameters (LSPs in our case).

On the one hand, a proposal by Kim et al. [9,4] called pseudocepstrum provides a straight and computationally efficient approximation to the LPCC parameterization. On the other, the LPCC parameterization can be computed in an exact and computationally more demanding way. Both approaches have been compared by the previous authors proving comparable performances considering the quantization errors introduced by a speech codec. In this paper, we compare their robustness to both the speech coding stage distortion and the impairments due to the IP transmission channel. In particular, we have tested both parameterizations in several simulated VoIP scenarios using two codecs (G.723.1 and G.729) and a wide range of Packet Loss Rates (PLRs) and Mean Burst Lengths (MBLs). This realistic testing environment adds to the analysis of the LSP quantization effects of [4] an evaluation of the influences of the whole codec process (for example on the energy parameter extraction or the frame rate provided) and the network distortions. This allows us to discuss when the approximation given by pseudocepstrum is advantageous and when, on the contrary, the exact LSP to MFCC conversion is preferable.

Finally, though not considered in this work, it is worth mentioning an alternative method for avoiding both the coding and decoding stage called Distributed Speech Recognition (DSR), which consists of a standard protocol for sending a specific type of ASR parameterization extracted at the user-end, instead of the coded version of the whole speech signal [6]. This is a very convenient alternative in terms of ASR performance, requiring only that the user terminal implements the standard parameterization defined in the DSR protocol. However, the widespread availability of standard speech coders allows the application of the bitstream-based approaches we analyze in this paper to be used when both the speech waveform and recognition are needed (in legal applications, for recording purposes, etc.)

In next Section we introduce the compared methods for obtaining MFCC-type representations from LSP coefficients followed by a discussion of their computational complexity in Section 3. Then, in Section 4, we report the experiments carried out and comment on the results. Finally, Section 5 summarizes our conclusions and outlines future work.

## 2 Bitstream-based parameterizations for ASR

When the speech signal has undergone a coding process, the first stage of a conventional ASR front-end is to decode it so that cepstral coefficients can

be subsequently extracted. The goal of this type of analysis is to obtain the spectral envelope of the speech signal, $H(\Omega)$, which is the most relevant information for ASR. This method (we will refer to as (classical) MFCC) separates $H(\Omega)$ from the excitation spectrum by liftering in the cepstral domain (also called homomorphic deconvolution). The cepstra vector so obtained will be denoted $\mathbf{c}_H^{(L)}$, where $L$ is the number of coefficients or, equivalently, the length of the lifter pass-band. Furthermore, if desired, a mel-scaled version, $\mathbf{mfc}_H^{(L)}$, can be easily computed by applying the corresponding weighting filter-bank in the frequency domain before computing the cepstral coefficients.

An alternative method for obtaining the spectral envelope of a speech signal is to use Linear Prediction (LP). The ASR parameterization obtained from the LP spectrum, $H_{LP}(\Omega)$, will be referred to as LPCC (Linear Prediction Cepstral Coefficients) or LP-MFCC (Linear Prediction Mel Frequency Cepstral Coefficients) if the mel scale is used.

When recognizing from the bitstream, the parameterization provided by the codec must be transformed into a more appropriate one, since recognizing directly from LSPs (or weighted versions) does not seem to be the best choice [4]. The problem can be stated as follows: given $\omega^{(P)}$, the $P$th-order LSP coefficient vector representing the all-pole synthesis filter for a particular speech frame, what is the best way to convert it, efficiently and reliably (from the point of view of robust speech recognition) into LPCC?

There are several ways to perform this conversion being the most obvious the transformation of LSP into LP coefficients to proceed with the well-known conversion from LP coefficients into cepstra. However, in this paper we focus on direct transformations from LSPs into LPCC (avoiding the intermediate computation of LP coefficients). Thus, two alternatives are compared, namely: the pseudocepstrum proposed by Kim et al. and an LSP to LPCC transformation suggested in this paper and based on the relationship between the LPC spectrum and the LSP parameters by N. Sugamura and F. Itakura [16].

### *2.1 From LSP to Pseudocepstra*

Kim et al. [9] derived an approximation of LPCC from the LSPs which they called 'pseudocepstrum', $\hat{\mathbf{c}}_{H_{LP}}^{(L)}$:

$$\hat{\mathbf{c}}_{H_{LP}}^{(l)} = \frac{1}{2l}\left(1+(-1)^{l}\right) + \frac{1}{l}\sum_{i=1}^{P}\cos\left(l\cdot\omega_i\right), \qquad 1 \le l \le L \tag{1}$$

where $l$ is the cepstral index and $\omega_i$ each one of the components of the LSP vector $\omega^{(P)}$. The complete deduction of this formula and the error or residue

that separates $\hat{\mathbf{c}}_{H_{LP}}^{(L)}$ from the exact LPCC can be found in [9]. This approximation also has the advantage that, since the LSPs are themselves frequencies, a mel-scaled version of these coefficients (*ps*-MFCC) can be easily obtained by direct weighting of the original LSPs:

$$\widehat{\mathbf{mfc}}_{H_{LP}}^{(l)} = \frac{1}{2l}\left(1+(-1)^{l}\right) + \frac{1}{n}\sum_{i=1}^{P}\cos\left(l\cdot \mathrm{mf}\omega_i\right), \qquad 1 \le l \le L \tag{2}$$

where $\{\mathrm{mf}\omega_i\}$ with $i = 1, \cdots, P$ are the mel-scaled LSP. Note that this would not be possible in the case of the LP coefficients though they convey the same information.

### *2.2 From LSP to LPCC*

The solution by Kim et al. is smart and computationally efficient. However, we must be aware of the fact that there is a difference (the residue) between pseudocepstra, $\hat{\mathbf{c}}_{H_{LP}}^{(L)}$, and the actual LPCC: $\mathbf{c}_{H_{LP}}^{(L)}$. This last observation brings about the following question: since the parameterization stage in ASR is not among the most computationally demanding subsystems, when is the use of this approximation advisable and when the computational gain obtained cannot compensate for the loss in ASR accuracy?

The LP spectrum can be obtained directly from LSP coefficients (see [16]). In particular, for an even $P$ (as it is the case in most speech coders), a discretized $N$-point spectrum, $|H_{LP}[k]|$, can be computed as follows:

$$|H_{LP}[k]| = \left(\frac{2^{P+2}}{\sin^2\left(\frac{\pi k}{N}\right)T_o[k] + \cos^2\left(\frac{\pi k}{N}\right)T_e[k]}\right)^{\frac{1}{2}} \qquad 0 \le k < N \tag{3}$$

where $T_o[k]$ and $T_e[k]$ are the products, expanded below, that account for the even and odd indexes of the $\omega^{(P)}$ vector, respectively:

$$\begin{aligned} T_o[k] &= \prod_{i=1}^{P/2}\left(\cos\left(\frac{2\pi k}{N}\right) - \cos\left(\omega_{2i}\right)\right)^2 \\ T_e[k] &= \prod_{i=1}^{P/2}\left(\cos\left(\frac{2\pi k}{N}\right) - \cos\left(\omega_{2i-1}\right)\right)^2 \end{aligned} \tag{4}$$

from which we can use conventional methods to obtain the cepstra (see for example [7]): an inverse DCT (Discrete Cosine Transform) of $\log|H_{LP}[k]|$ followed by a liftering stage leads to the desired $\mathbf{c}_{H_{LP}}^{(L)}$. Likewise, the corresponding mel-scaled coefficients, $\mathbf{mfc}_{H_{LP}}^{(L)}$ (i.e., LP-MFCC) result by applying the mel scaling to the frequency axis before computing the inverse DCT.

It is worth noting that, in contrast to pseudocepstrum, none of these transformations entails any approximation. As previously mentioned, it would also be possible to start converting LSP into LP coefficient and subsequently proceed in a similar way. Nevertheless, experimental results do not show any significant difference with respect to the above described and theoretically equivalent, LSP to cepstrum transformation. This fact makes it more advisable (from a computational point of view) to use this last transformation as it avoids the initial LSP to LP conversion step.

## 3 Computational Complexity Issues

We have compared the computational cost of a Matlab implementation of both $\mathbf{mfc}_{H_{LP}}^{(L)}$ and $\widehat{\mathbf{mfc}}_{H_{LP}}^{(L)}$. These implementations have been carefully designed to avoid any recalculations, precomputing parameter values whenever possible. By averaging 500 realizations of both alternatives, we have obtained that the mean time needed for the computation of mel-scaled pseudocepstrum is about 0.1 times that the one employed in the calculation of LP-MFCC.

These results show the superiority of pseudocesptrum in terms of computational efficiency. However, this gain must be considered in the context of the overall complexity of the ASR process. In particular, the LP-MFCC parameterization process represents for our ASR tasks (described later) between 10% and 15% of the whole ASR process. Therefore, the computational gain due to pseudocepstrum will be between 9% and 13.5%.

## 4 Experiments and Results

For the purpose of comparatively assessing the robustness of both pseudocepstrum and LPCC front-ends in a (to some extent) realistic scenario, we have tested them in a VoIP environment. This includes two standard speech codecs (G.723.1 and G.729) widely used in those types of networks and a simulation of the packet loss patterns inspired in real-traffic measurements [1,14].

### *4.1 Baseline Setup*

For our experiments, we have chosen a speaker-independent continuous speech recognition (CSR) task. We have used the well-known Resource Management RM1 Database [13], which has a 991-word vocabulary. The speaker-independent training corpus consists of 3,990 sentences pronounced by 109

speakers and the test set contains 1,200 sentences from 40 different speakers. This corresponds to a compilation of the first four official test sets. Originally, RM1 was recorded at 16 kHz in clean conditions; however, our experiments were performed using a (down-sampled) version at 8 kHz and clean conditions thus allowing us to concentrate exclusively on channel and codec distortions. We have employed context-dependent acoustic models; namely, three-mixture cross-word triphones. A standard word-pair grammar was used as the language model and HTK has been used as the implementation tool [18].

In order to state the statistical significance of our experiments we have calculated confidence intervals (for a confidence of 95%) using the formula [17, pp. 407-408]:

$$\frac{b}{2} = 1.96\sqrt{\frac{p\,(100-p))}{n}} \tag{5}$$

where $p$ is the recognition rate (%) for the described task and $n$ is the number of examples to be recognized (in our case, 10,288 words). Thus, any recognition rate is presented as belonging to the band $[p - b/2, p + b/2]$.

### *4.2 Parameterization*

We have employed a 12th dimensional cepstral vector ($L = 12$) plus an energy value and their corresponding delta parameters in all of the parameterizations evaluated; thus, feature vectors comprise 26 components. The number of LSP provided by both G.723.1 and G.729 codecs is $p = 10$.

When using the bitstream-based approach both the frame rate (FR), (i.e., the time interval between two consecutive feature vectors), and the energy estimation procedures have proven to be of great importance [15,7]. With respect to the FR, only the G.723.1 codec needs an adaptation since it provides a frame period of 30 ms (not suitable for speech recognition) that contrasts with the 10 ms given by G.729. Therefore, once the LSP parameters have been extracted from the G.723.1 bitstream, an interpolation stage is applied to establish the appropriate FR. More precisely, a band-limited interpolation FIR filter is applied over the time sequence of each component of the feature vector reducing the frame period from 30 ms to 10 ms (the period usually employed by conventional ASR front-ends). This interpolation filter uses the four nearest (2 from each side) samples. It is important to note, however, that this filter does not cause any additional delay, since we have already admitted this delay for the computation of the dynamic parameters.

In addition an energy parameter has been computed in both bitstream-based parameterizations from a subset of the parameters directly extracted from the bitstream (see [15] for a detailed description).

Finally, the LSP are transformed into cepstrum-type parameters. As previously mentioned, in this paper we compare two transformations. We will use mel-scaled versions of them and refer to them as ps-MFCC (mel-scaled pseudo-cepstrum) and LP-MFCC.

### *4.3 Simulated IP Channels*

To simulate packet loss in IP networks, we have employed a two-state Gilbert model [8]. We have considered several IP channels, with different Packet Loss Rates (PLR) and Mean Burst Lengths (MBL). Channels A-F were already used by the authors in [15], while the channels G-J are those used by Boulis et al. in [2].

The number of speech frames that should be included in an IP packet is a compromise that should balance the resources employed in the packet headers and the actual transmitted information (i.e. the coded speech): a very small payload would incur in what is known as an overhead problem where the efficiency of the network is compromised. In particular, with the G.723.1 codec we have decided to include just one frame per packet given that each of them represents 30 ms of speech. However, with G.729, whose frame rate is 10 ms, we have located three frames per packet to avoid overhead and to allow for a better comparison between the results of the two codecs.

The values of PLR and MBL displayed in Figures 1 and 2 are measures empirically obtained from the application of the Gilbert model to the coded and packetised RM1 database. Therefore, these figures can vary slightly from the theoretical ones or even from the ones obtained in different trials and would converge for a database of infinite duration. However, they have been kept identical for all the parameterizations compared and are exact indicators of the damage done to the test feature vectors.

Finally, it is worth mentioning that the error concealment mechanisms considered in recommendations G.723.1 and G.729 are also activated for all the compared options.

### *4.4 Comparison of Front-Ends*

Figures 1 and 2 display the recognition results for channels A to J. The first observation we can make is that LP-MFCC is superior to mel-scaled pseudo-cepstrum for all the considered cases. Furthermore, the improvement achieved is greater as the channel conditions worsen. In other words, as the rate of lost frames and mean burst length increases, the residue of the approximation

in the pseudocepstrum approach becomes more relevant. Besides, the gain in accuracy obtained by the LP-MFCC parameterization seems to be more correlated with the MBL showing that the approximation error becomes more relevant when the number of frames consecutively missing increases. Nevertheless, the differences in performance are not significant in most cases which can make the ps-MFCC approach preferable in certain applications due to its computational advantages.

Besides, if we compare the results for the two codecs we realize that the previous conclusions are consistent and hold for both of them even though they are quite different in bit and frame rates. However, the overall performance of G.729 is moderately better (for both LP and ps-MFCC parameterizations) and the gains obtained by the LP-MFCC parameterization slightly smaller.

## 5 Conclusions and Further Work

The so-called bitstream-based ASR approach has proved to be one of the best alternatives when spoken interfaces for VoIP-based applications are considered. In this context, the speech signal is encoded by means of a standard codec. Bitstream-based ASR systems aim at extracting the parameterization directly from the bitstream instead of decoding the speech signal and proceeding as usual.

Motivated by the fact that most of the speech coders employed for Voice over IP communications encode the spectral envelope of the speech signal in the form of LSP coefficients, we have compared, for several IP channel conditions, the robustness against coding distortion and packet losses of two parameterization methods which transform LSPs into cepstrum-type parameters (more appropriate for recognition purposes). These methods are the 'pseudocepstrum' approximation (ps-MFCC) and an LP-based cepstrum (LP-MFCC).

Summing up, ps-MFCC demands significantly less computational effort, but its performance degrades in comparison with that of LP-MFCC, especially as the network conditions worsen. Therefore, ps-MFCC is an interesting alternative when the computational resources are low and PLR and MBL can be maintained under certain levels, while LP-MFCC is recommended when favorable network conditions cannot be guaranteed.

Besides, due to the importance of the communications channels in current speech technologies, we find interesting the extension of this type of analysis to the mobile phone communication environment. In this context, bitstream-based ASR has also proven to be a promising alternative [7,10] and the distortion due to transmission errors deserves a specific analysis.

## References


[1] M.S. Borella, Measurement and interpretation of Internet packet loss, *Journal of Communications and Networking*, vol. **2**, no. **2**, 2000, pp. 93–102.

[2] C. Boulis, M. Ostendorf, E.A. Riskin and S. Otterson, Graceful degradation of speech recognition performance over packet-erasure networks, *IEEE Trans. on Speech and Audio Processing*, vol. **10**, no. **8**, 2000, pp. 580–590.

[3] S.H. Choi, H.K. Kim and H.S. Lee, LSP weighting functions based on spectral sensitivity mel-frequency warping for speech recognition in digital communications, *Proc. ICASSP*, vol. **1**, 1999, pp. 401–404.

[4] S.H. Choi, H.K. Kim and H.S. Lee, Speech recognition using quantized LSP parameters and their transformations in digital communications, *Speech Communication*, vol. **30**, no. **4**, 2000, pp. 223–233.

[5] S.H. Choi, H.K. Kim, H.S. Lee and R.M. Gray, Speech recognition method using quantised LSP parameters in CELP-type coders, *Electronics Letters*, vol. **34**, no. **2**, 1998, pp. 156–157.

[6] ETSI Speech processing Transmission and Quality aspects (STQ), Distributed speech recognition (DSR); Front-end feature extraction algorithm; Compression algorithms, *ES 201 108*, 2000.

[7] A. Gallardo-Antolín, C. Peláez-Moreno and F. Díaz-de-María, Recognizing GSM digital speech, *IEEE Trans. on Speech and Audio Processing* (to appear).

[8] L.N. Kanal and A.R.K. Sastry, Models for channels with memory and their applications to error control, *Proc. of the IEEE*, 1978, pp. 724–744.

[9] H.K Kim, S.H. Choi and H.S. Lee, On approximating Line Spectral Frequencies to LPC cepstral coefficients, *IEEE Trans. on Speech and Audio Processing*, vol. **8**, no. **2**, 2000, pp. 195–199.

[10] H.K. Kim, R.V. Cox and R.C. Rose, Performance improvement of a bitstream-based front-end for wireless speech recognition in adverse environments, *IEEE Trans. on Speech and Audio Processing*, vol. **10**, no. **8**, 2002, pp. 591–604.

[11] A.M. Kondoz, *Digital Speech: Coding for Low Bit Rate Communication Systems*, John Wiley & Sons, 1996.

[12] B.T. Lilly and K.K. Paliwal, Effect of speech coders on speech recognition performance, *Proc. ICSLP*, Philadelphia, USA, 1996, no. **4**, pp. 2344–2347.

[13] National Institute of Standards and Technology (NIST) (distributor), *The Resource Management corpus part 1 (RM1)*, 1992.

[14] V. Paxon, *Measurements and Analysis of End-to-End Internet Dynamics*, PhD Thesis, Berkeley, University of California, 1997.

[15] C. Peláez-Moreno, A. Gallardo-Antolín and F. Díaz-de-María, Recognizing voice over IP networks: a robust front-end for speech recognition on the WWW, *IEEE Trans. on Multimedia*, vol. **3**, no. **2**, 2001, pp. 209–218.

[16] N. Sugamura and F. Itakura, Speech analysis and synthesis methods developed at ECL in NTT -from LPC to LSP-, *Speech Communication*, vol. **5**, no. **2**, 1986, pp. 199–215.

[17] N.A. Weiss and M.J. Hasset, *Introductory Statistics*, Addison-Wesley, 1993.

[18] S. Young et al., *HTK-Hidden Markov Model Toolkit (ver. 3.0)*, Cambridge University, 2000.

**List of Figures**

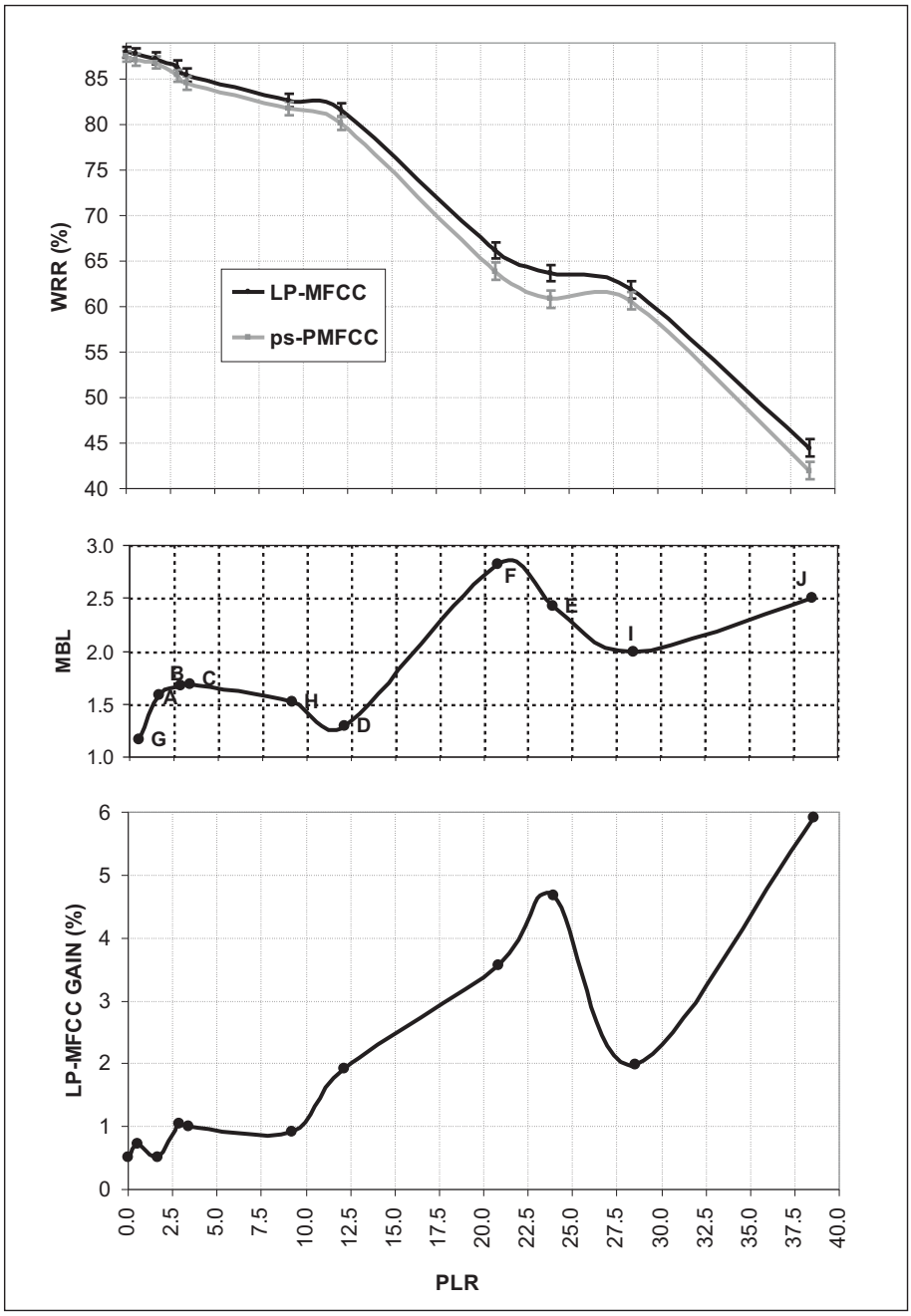


Fig. 1. Comparative ASR results for the two considered parameterisations and different PLRs for G.723.1: absolute word recognition results including confidence intervals (upper part), mean burst lengths (MBL) measured in number of packets (center) and relative gain obtained by using LP-MFCC instead of ps-MFCC (lower part).

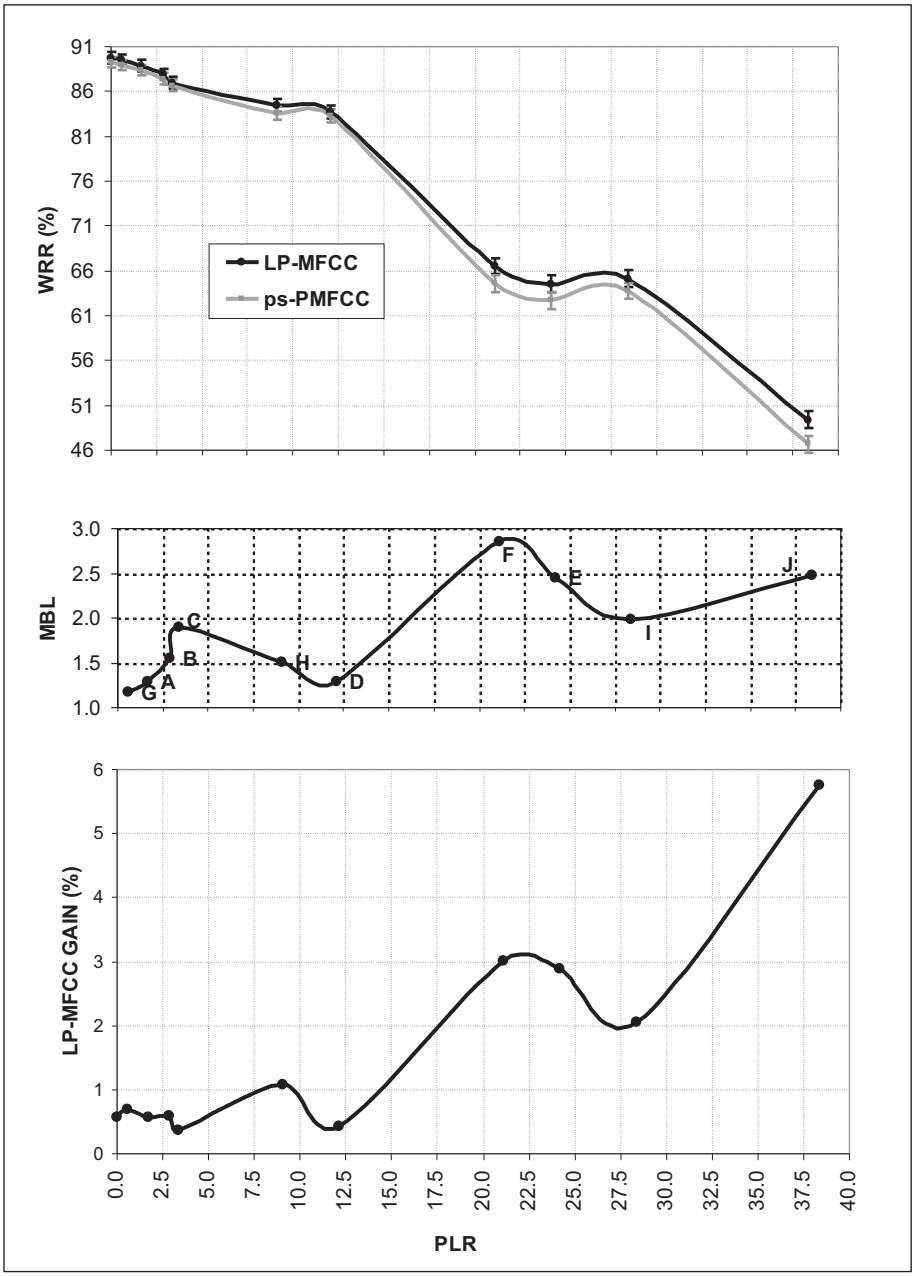


Fig. 2. Comparative ASR results for the two considered parameterisations and different PLRs for G.729 using three frames per packet: absolute word recognition results including confidence intervals (upper part), mean burst lengths (MBL) measured in number of packets (center) and relative gain obtained by using LP-MFCC instead of ps-MFCC (lower part).